

Singapore Management University Institutional Knowledge at Singapore Management University

Research Collection School Of Information Systems

School of Information Systems

1-2009

Quality-aware collaborative Question Answering: Methods and evaluation

Maggy Anastasia SURYANTO

Singapore Management University

Ee Peng LIM

Singapore Management University, eplim@smu.edu.sg

Aixin SUN


Nanyang Technological University

Roger Hsiang-Li CHIANG

University of Cincinnati

DOI: <https://doi.org/10.1145/1498759.1498820>

Follow this and additional works at: https://ink.library.smu.edu.sg/sis_research

 Part of the [Databases and Information Systems Commons](#), and the [Numerical Analysis and Scientific Computing Commons](#)

Citation

SURYANTO, Maggy Anastasia; LIM, Ee Peng; SUN, Aixin; and CHIANG, Roger Hsiang-Li. Quality-aware collaborative Question Answering: Methods and evaluation. (2009). *WSDM '09: Proceedings of the Second ACM International Conference on Web Search and Data Mining*. 142-151. Research Collection School Of Information Systems.

Available at: https://ink.library.smu.edu.sg/sis_research/451

This Conference Proceeding Article is brought to you for free and open access by the School of Information Systems at Institutional Knowledge at Singapore Management University. It has been accepted for inclusion in Research Collection School Of Information Systems by an authorized administrator of Institutional Knowledge at Singapore Management University. For more information, please email libIR@smu.edu.sg.

Quality-Aware Collaborative Question Answering: Methods and Evaluation

Maggy Anastasia Suryanto
School of Computer Engineering
Nanyang Technological University
magg0002@ntu.edu.sg

Aixin Sun
School of Computer Engineering
Nanyang Technological University
axsun@ntu.edu.sg

Ee-Peng Lim
School of Information Systems
Singapore Management University
eplim@smu.edu.sg

Roger H. L. Chiang
Information Systems Dept, College of Business
University of Cincinnati
roger.chiang@uc.edu

ABSTRACT

Community Question Answering (QA) portals contain questions and answers contributed by hundreds of millions of users. These databases of questions and answers are of great value if they can be used directly to answer questions from any user. In this research, we address this collaborative QA task by drawing knowledge from the crowds in community QA portals such as Yahoo! Answers. Despite their popularity, it is well known that answers in community QA portals have unequal quality. We therefore propose a quality-aware framework to design methods that select answers from a community QA portal considering answer quality in addition to answer relevance. Besides using answer features for determining answer quality, we introduce several other quality-aware QA methods using answer quality derived from the expertise of answerers. Such expertise can be question independent or question dependent. We evaluate our proposed methods using a database of 95K questions and 537K answers obtained from Yahoo! Answers. Our experiments have shown that answer quality can improve QA performance significantly. Furthermore, question dependent expertise based methods are shown to outperform methods using answer features only. It is also found that there are also good answers not among the best answers identified by Yahoo! Answers users.

Categories and Subject Descriptors

H.3.4 [Information Storage and Retrieval]: Systems and Software—*question-answering (fact retrieval) systems*

General Terms

Experimentation

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

WSDM '09 Barcelona, Spain

Copyright 2009 ACM 978-1-60558-390-7 ...\$5.00.

Keywords

Question answering, expertise, answer quality

1. INTRODUCTION

1.1 Motivation

In this paper, we address the Collaborative QA task that involves answering questions using answers available from a community QA portal. A community QA portal usually consists of a large group of users contributing questions and answers online. Examples of such portals include Yahoo! Answer¹, answerbag², wondir³, Naver⁴, etc. Among them, Yahoo! Answers is reported to have millions of questions, answers, and users. Each community QA portal usually has a large pool of questions and answers for user browsing and searching. Users who have questions (especially the popular ones) may therefore find ready answers in a community QA portal. This can save them much time and efforts searching or researching for answers.

Collaborative QA, i.e., finding good answers to a question using community QA portals is nevertheless a challenging task. Given a user query, the search engine of a typical community QA portal matches it with the questions from its database, and returns the questions ranked by relevance. It is usually the user's responsibility to manually select the appropriate returned questions and determine the most appropriate answer(s).

The above search approach unfortunately suffers from *answer quality problem*[13, 24, 4], i.e., the answers in the system may be irrelevant and/or poorly written even if their associated questions are relevant. Users give poor quality answers due to several reasons including limited knowledge about the question domain, bad intentions (e.g., spam, making fun of others, etc.), limited time to prepare good answers, etc.

Some community QA portals have implemented user feedback as a counter-measure to overcome the quality problem. For example, users may report abusive answers or questions (such as spams, adult content or other abusive contents),

¹<http://answers.yahoo.com>

²<http://www.answerbag.com>

³<http://www.wondir.com>

⁴<http://www.naver.com>

and rate other users' answers. In addition, a question's asker may select one best answer out of all answers posted, or allow other users to vote the best answer to the question.

However, the existing search engines of community QA portals do not effectively utilize user feedback. Even so, they are not likely to perform well if the feedback is utilized in a naive manner because user feedback are voluntary, subjective and not reliable. Recently, Gyöngyi et al [9] reported that in 10 months worth of Yahoo! Answers data, over 30 % of best answer selection were affected by self votes which are votes cast for an answer by the user who provides that answer.

1.2 Research Objectives and Contributions

In this research, we aim to introduce different methods to automatically find good answers for a user given questions from a community QA portal. By directly returning answers to a user, we hope to reduce the efforts required to locate good answers. This hopefully will also reduce the need to post duplicate questions in the community QA portal.

One obvious method to find good answers is to return the best answers of top ranked questions returned by the portal's search engine. The method assumes that the search engine ranks returned questions by relevance and each question has a best answer voted by its asker or other users. This assumption holds in most community QA portal's search engines (e.g., Yahoo! Answers, Answerbag, etc). The method is however likely to suffer from quality problems and we shall use it as a baseline (see BasicYA in Section 5.2) for comparing with other proposed methods.

To solve the answer quality problems, we would like to consider (a) *answer features*, and (b) *user expertise* of answerers. In particular, we focus on using user expertise to derive answer quality. As our experiment results show later in the paper, expertise based QA methods yield better answer quality than answer feature based methods.

In the following, we summarize our objectives and contributions in this paper to address the answer quality issues when conducting question answering on a community QA portal:

- We introduce a quality-aware QA framework that considers both answer relevance and quality in selecting answers to be returned. This framework can be used to derive different QA methods by adopting different relevance and quality scoring functions.
- We develop several QA methods (namely, EXHITS, EXHITS_QD, EX_QD and EX_QD') that consider answerer expertise to determine answer quality. These methods consider the expertise of a user in both asking and answering questions. Since an answerer's expertise may be closely associated with the question topics, EXHITS_QD, EX_QD and EX_QD' adopt question dependent expertise in measuring answer quality. Finally, we also incorporate the options of using all answers or best answers only as candidate answers into our methods.
- We have conducted experiments to evaluate our proposed QA methods on a set of test questions and a large user labeled collection of 95K questions and 537K answers from Yahoo! Answers. We evaluate both the overall and quality performance of the QA methods.

It has been found that quality-aware methods can improve both quality and overall performance. Among them, the methods EX_QD and EX_QD' using question dependent answerer expertise have the best performance.

Although our study focuses on Yahoo! Answers, the most popular community QA portal, most of the ideas should be applicable to other community QA portals.

1.3 Paper Organization

The rest of this paper is structured as follows. The related work and our proposed QA framework are covered in Sections 2 and 3 respectively. We present our proposed expertise based QA methods in Section 4. Sections 5 and 6 describe the experimental setup and results respectively. Finally, we conclude our work in Section 7.

2. RELATED WORK

There has been extensive research on question answering (QA) since the task was introduced in late 90's [8, 10, 20]. The traditional QA research focuses on searching and extracting answers from a static collection of text segments or web data [2, 19]. The traditional QA solutions are very much content based. There was some QA research for FAQ data maintained by knowledgeable users [5, 22]. Our work is different since we deal with an archive of answers from users of a community QA portal that suffers from unequal answer quality problem.

Previous work on collaborative QA focused on retrieving the most relevant questions given a new user question. Jeon et al [12] proposed using machine translation models to find semantically similar existing questions from a community QA portal. However, they did not take quality of answers into consideration.

Jeon et al [13] subsequently addressed the answer quality problem in a community QA portal using a classification approach. He derived a set of non-textual answer features, such as answer length, answerer's number of answers, questions and best answers made so far, answer rating, etc, for determining answer quality. We borrow some of their idea but also explore using user expertise to measure answer quality.

In the context of community QA portals, Jurczyk and Agichtein [15, 16] adopted HITS algorithm [17] to measure the authority (a form of expertise) of users in a bipartite network where an asker is linked to an answerer when a question posted by the former has been answered by the latter. They reported that the obtained authority score is a better measure of expertise than simply counting the number of answers an answerer has given. Agichtein et al [1] further derived multiple user interaction graphs for different kinds of relationships in a community QA portal, such as asking-answering, selecting best answers, abuse reporting and answer voting/rating. They then computed the authority and hub values of each user in the graphs and used them as answer features in their regression technique to predict answer quality. Their work is very different to ours as they did not address the entire collaborative QA task. Besides, their user expertise definition is question independent and is reported to give insignificant performance improvement as compared to the answer textual features they used [1]. Our user expertise definitions however can be question dependent (see Section 4) which, according to our experiments, yielded more significant performance improvement (see Section 6.2).

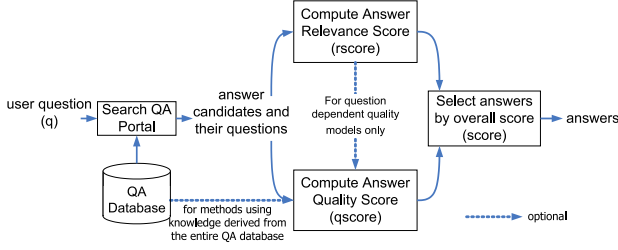


Figure 1: Proposed framework.

The work by Bian et al [4] is the most related to ours. They proposed to solve collaborative QA by considering both answer quality and relevance. Similar to earlier work on quality measurement, Bian et al also used content-based quality answers without considering user expertise. Their work was also confined to answer factoid questions and assumed the provision of large relevance labeled data.

Liu and Croft[25] addressed the experts finding problem in a community QA portal where experts of a given topic are to be found among the answerers in the system. They characterized the expertise of an answerer using a profile derived from combining previously answered questions. They addressed the problem as an IR problem where a given question is viewed as query and the expert profiles as documents. Expert finding for other types of systems ([3, 7, 6, 23] for enterprise systems consisting of technical reports and emails, [27] for Internet forums, etc.), has also been studied extensively.

3. QUALITY-AWARE FRAMEWORK

Our QA solution framework is depicted in Figure 1. It first involves sending the user question (q) to the QA portal’s search engine which returns a set of questions and their answers. The next step is to assign a relevance score ($rscore$) and a quality score ($qscore$) to each returned answer. Here, an answer’s relevance to the user question can be determined by the relevance of the question associated with the answer. In other words, all answers of the same question share the same relevance score when evaluated against a user given question. The quality of an answer can be determined by the answer content as well as from the knowledge (e.g., expertise) about the user contributing the answer. This knowledge may be computed from the entire database of questions and answers. Finally, the answers are ranked by combining the relevance and quality scores.

Our framework does not consider the question quality directly as it appears to be a relatively minor problem compared to answer relevance and quality in most community QA portals (at least for Yahoo! Answers). This is perhaps due to incentive schemes that try to weed out poor quality questions by penalizing users for posting questions that do not attract answers or receive negative responses from users.

In the above framework, answer quality can be determined by (a) *answer features*, and (b) *user expertise*. Jeon et al proposed using a set of non-textual features in answers to judge their quality [13]. We borrow their method for measuring answer in a quality-aware QA method called NT. We will present the NT method together with the other baseline methods in Section 5.2.

Answer quality can be derived using different expertise models. In this paper, we would like to explore several of

Table 1: Expertise based methods

Method	Question Dependency	Peer Expertise Dependency	Remark on $e_{ask}(u, q)$
EXHITS	independent	dependent	N/A
EXHITS_QD	dependent	dependent	considers answer feature weight
EX_QD	dependent	independent	considers answer feature weight
EX_QD'	dependent	independent	doesn't consider answer feature weight

them, including question independent and dependent expertise, and both answerer’s asking and answering expertise.

When returning answers to the user, one can choose to return all answers or best answers, ranked by overall scores. This two answer options may potentially affect the QA performance. For the non-quality aware methods, only the best answers are returned since all answers of the same question share the overall score (i.e., $rscore$). Hence, the all answer option is not applicable to quality-aware methods.

4. EXPERTISE BASED METHODS

Based on our quality-aware QA framework, our aim is to select the answers ranked by their overall scores. Given a new question q , we determine the overall score of an answer a (i.e., $score(q, a)$) using Equation 1, i.e., direct product of the relevance score ($rscore(q, a)$) and the quality score ($qscore_{\langle model \rangle}([q, a])$) of a . Depending on the quality model used $\langle model \rangle$, a different quality-aware QA method is derived.

$$score(a) = rscore(q, a) \cdot qscore_{\langle model \rangle}([q, a]) \quad (1)$$

In this section, we associate the quality of an answer with the expertise of its answerer. We consider two expertise models, namely *question dependent* and *question independent*. The former assumes that answerer’s expertise is independent of the topic of the question q , while the latter assumes otherwise. An answerer’s expertise also consist of *asking expertise*, the ability in asking good questions, and *answering expertise*, the ability in giving good answers.

The expertise of a user can be inferred by his/her experience and peer (other users’) acknowledgement. The previous questions and answers made are part of the past experience. We use asking and replying exchanges with the other users as a form of acknowledgement among users. The better expertise answerers giving answers to a user’s question, the better asker the user is. The better expertise askers whose questions are answered by a user, the better answerer the user is. We shall refer this mutual relationship between asking and answering as *peer expertise dependency*.

With the above considerations, we therefore arrive at the following four quality-aware expertise based QA methods: (a) EXHITS, (b) EXHITS_QD, (c) EX_QD, and (d) EX_QD' as summarized in Table 1. All the four methods derive users’ asking and answering expertise from their past questions and answers made. Before we elaborate on them in Sections 4.1 and 4.2, we first define the symbols common to these methods. As shown in Figure 2, we use $u_i \rightarrow q_k$ to represent u_i posting question q_k and $u_i \leftarrow a_{kl}$ to represent u_i answering q_k with answer a_{kl} . Note that a user can be both an asker and an answerer (e.g. u_1) and each question has only one asker.

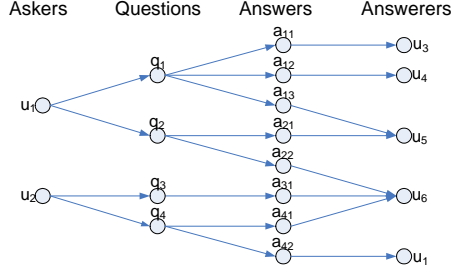


Figure 2: Users, questions and answers.

4.1 Question Independent Expertise

EXHITS refers to the quality-aware expertise based QA method that uses $qscore_exhits(a)$ as the quality score to compute overall score of an answer a as given in Equation 2. The quality score of a is derived from the asking expertise (e_ask) and answering expertise (e_ans) of its answerer.

$$qscore_exhits(a) = \sigma \cdot e_ans(u_i) + (1 - \sigma) \cdot e_ask(u_i) \quad (2)$$

where u_i is the answerer of a ($u_i \leftarrow a_{jl}$) and σ ($0 \leq \sigma \leq 1$) is the parameter that controls the importance of answering expertise with respect to asking expertise of u_i .

EXHITS uses the peer expertise dependency and computes both asking expertise and answering expertise using the HITS model on the entire question and answer database as proposed by Jurczyk and Agichtein [15]. Here, $e_ask(u_i)$ and $e_ans(u_i)$ are very much the hub and authority of the user u_i . As shown in Equation 3, the asking expertise (answering expertise) of u_i is derived from the answering expertise (asking expertise) of users answering questions posted (posting questions answered) by u_i , i.e., $\{q_k\}$ ($\{a_{kl}\}$). EXHITS then determines the quality of an answer and combines it with relevance score to obtain the final answer score.

$$\begin{aligned} e_ask(u_i) &= \sum_{u_i \rightarrow q_k} \sum_{u_j \leftarrow a_{kl}} e_ans(u_j) \\ e_ans(u_i) &= \sum_{u_i \leftarrow a_{kl}, u_j \rightarrow q_k} e_ask(u_j) \end{aligned} \quad (3)$$

4.2 Question Dependent Expertise

Instead of assuming each user having the same level of expertise for different topics, the question dependent expertise models calibrate different levels of expertise for the same user when he or she answers different question topics. This new assumption sounds logical since users usually have diverse background and experience.

In this section, we propose three question dependent expertise based QA methods, namely, EXHITS_QD, EX_QD and EX_QD'. These methods compute the quality of an answer using the formula presented in Equation 4. Unlike Equation 2, the new equation has incorporated q into the quality formulas.

$$qscore_model(q, a) = \sigma \cdot e_ask(u_i, q) + (1 - \sigma) \cdot e_ans(u_i, q) \quad (4)$$

where u_i is the answerer of answer a and σ is the parameter that controls the importance of answerer expertise relative to asker expertise.

EXHITS_QD is the question dependent version of EXHITS method. We denote the asking expertise of u_i in ask-

ing question q by $e_ask(u_i, q)$, and answering expertise of u_i in answering user question q by $e_ans(u_i, q)$. As shown in Equation 5, $e_ask(u_i, q)$ is the aggregation of the answering expertise of those users answering the questions posted by u_i , $\{q_k\}$, weighted by the relevance ($rscore(q, q_k)$) of the posted questions q_k with respect to q , and non-expertise based quality scores (denoted by w_{kl} 's) of the answers $\{a_{kl}\}$ given. We expect $e_ask(u_i, q)$ to be high when there are users with good answering expertise answering u_i 's questions that are relevant to q . Similarly, the answering expertise of a user u_i is the aggregation of the asker expertise of their corresponding askers weighted by the relevance of questions answered and non-expertise based quality scores of answers they provide.

The non-expertise quality score of an answer serves as a preliminary measure of its goodness. In our experiments, we set w_{kl} to be $qscore_nt(a_{kl})$ that is derived from non-textual features of a_{kl} as defined in Section 5.2.

$$\begin{aligned} e_ask(u_i, q) &= \sum_{u_i \rightarrow q_k} rscore(q, q_k) \cdot \sum_{u_j \leftarrow a_{kl}} w_{kl} \cdot e_ans(u_j, q) \\ e_ans(u_i, q) &= \sum_{u_i \leftarrow a_{kl}, u_j \rightarrow q_k} w_{kl} \cdot rscore(q, q_k) \cdot e_ask(u_j, q) \end{aligned} \quad (5)$$

EX_QD is a non-peer expertise dependent counterpart of EXHITS_QD. This model assumes the following:

- expert askers of question q are those users who have asked many other questions that are related to q and attracted many answerers to answer their questions with good answers
- expert answerers of question q are those users who provide good answers to many other questions that are related to q

Different from EXHITS_QD, EX_QD assumes the asking and answering expertise of a user to be independent from those of other users. Thus, the asking expertise and answering expertise are computed without recursion as shown in Equation 6.

$$\begin{aligned} e_ask(u_i, q) &= \sum_{u_i \rightarrow q_k} rscore(q, q_k) \cdot \sum_{u_j \leftarrow a_{kl}} w_{kl} \\ e_ans(u_i, q) &= \sum_{u_i \leftarrow a_{kl}, u_j \rightarrow q_k} w_{kl} \cdot rscore(q, q_k) \end{aligned} \quad (6)$$

EX_QD', the last expertise based method whose asking expertise is a slightly modification of that of EX_QD. It assumes the following:

- expert askers of question q are those users who have asked many other questions that are related to q
- expert answerers of question q are those users who provide good answers to many other questions that are related to q

Different from EX_QD's assumption on answering expertise, EX_QD' considers the past questions only when measuring asking expertise. Thus, the asking and answering expertise are computed as shown in Equation 7.

$$\begin{aligned} e_ask(u_i, q) &= \sum_{u_i \rightarrow q_k} rscore(q, q_k) \\ e_ans(u_i, q) &= \sum_{u_i \leftarrow a_{kl}, u_j \rightarrow q_k} w_{kl} \cdot rscore(q, q_k) \end{aligned} \quad (7)$$

5. EXPERIMENTAL SETUP

We conducted several experiments to evaluate the performance of our proposed methods so as to answer the following research questions:

- Does quality affect the overall collaborative QA performance? If so, can the quality-aware methods help to improve the QA performance?
- What is the performance of expertise based methods (EXHITS, EXHITS_QD, EX_QD, EX_QD') compared to that of the answer feature based method (denoted by NT)?
- The quality-aware methods may consider all answers or best answers only as answer candidates. Which is a better answer option?

5.1 Answer Relevance Models

We consider two answer relevance models in our experiments. The first is the answer ranking by Yahoo! Answers search engine. The second is the query likelihood retrieval model[21].

Other than the above mentioned two relevance models, we have tested other simple relevance models including VSM and Okapi BM25. From this experiment, the query likelihood and Yahoo! Answers retrieval models performed slightly better than the other experimented models.

We are aware that there are more sophisticated relevance models that use semantic knowledge to address the semantic term mismatch problem in retrieval. Since at this moment our current research focuses on comparing different quality aware methods and their impact on overall performance, we choose to use the simple relevance model. Exploring other relevance models for collaborative QA is an interesting topic which can be investigated in the future work.

Given a user question, Jeon [12] and Jijkoun [14] reported that measuring the relevance of an answer using the associated question is better than using the answer itself. Thus, in the query likelihood retrieval model, the relevance of an answer a_{kl} to a user question q is given by the probability of generating q from the *existing question language model* of q_k , $P(q|q_k)$.

$$\begin{aligned} rscore(q, a_{kl}) &= rscore(q, q_k) \\ &= P(q_k|q) = P(q_k)P(q|q_k)/P(q) \end{aligned} \quad (8)$$

We shall omit $P(q)$ in our computation since it does not affect ranking. We further assume uniform existing-question prior $P(q_k)$. For the existing question model, unigram language model is used to estimate $P(q|q_k)$.

$$P(q|q_k) = \prod_{w \in q} P(w|q_k) \quad (9)$$

To estimate more accurate existing question model, we applied Jelinek-Mercer background smoothing[26]:

$$P(w|q_k) = (1 - \lambda)Pml(w|q_k) + \lambda \cdot Pml(w|C) \quad (10)$$

where $Pml(w|q_k)$ is the maximum likelihood estimate of generating word w from question q_k . $Pml(w|C)$ is the maximum likelihood estimate of generating word w from the collection C that consists of all answers and questions in the dataset. λ is a smoothing parameter. In our experiment, we set $\lambda = 0.2$ that has been reported to perform well for short query titles [26].

5.2 Baseline Methods

We used three methods as baselines: **BasicYA**, **BasicQL** and **NT**. BasicYA and BasicQL rank questions based on relevance only and returns the best answers of top ranked questions without examining their quality. NT on the other hand use a prediction model to determine the quality of answers.

BasicYA uses the relevance ranking as implemented by Yahoo! Answers. Yahoo! Answers allows query to be processed with the following search options: best answers (**b**), question subject/title + content/description (**s+c**) and all (**s+c+b**). We experimented two variants of BasicYA, which are BasicYA(**s+c**) and BasicYA(**s+c+b**). The best answers (**b**) search option is not considered. We expect this option to perform poorly since it measures the relevance of the answer content instead of the question content.

BasicQL adopts query likelihood retrieval model to score the relevance of an answer as described in Section 5.1. We experimented two variants of BasicQL, which are BasicQL(**s**) and BasicQL(**s+c**).

NT. Jeon et al.[13] proposed to use Maximum Entropy approach [18] to build a stochastic process to predict the quality of an answer using its features. While their main objective was to show that the predictor has the ability to distinguish good answers from bad ones, they reported that their work can be used for finding good answers given a user question.

In our experiment, we used 8 of the 13 features proposed in Jeon et al's work and one additional feature marked by ‡ sign. The other 5 features were not used because they are either not available or not provided by Yahoo! Answers (such as click, copy and print counts). The 9 answer features we used are:

1. Proportion of best answers given by the answerer
2. Answer length
3. # stars (*) (One to five stars) given by the asker to the answer should it be selected as the best answer. Otherwise a zero value is assigned
4. # answers the answerer has provided so far
5. # categories that the answerer is declared the top contributor at (cap at 3)
6. # times the answer is recommended by other users
7. # times the answer is dis-recommended by other users
8. # answers for the question associated with the answer
9. ‡ # points that the answerer receives from answering, giving best answers, voting and signing in

We applied feature conversion on the non-monotonic features using Kernel Density Estimation[11] as proposed by the original work. Due to space limitation, we shall leave out the detail.

A training set to be elaborated in Sections 5.3 and 5.4, was constructed. The training set consisted of pairs of (a, y) , where a is an answer and y is a label, $y \in \{good, bad\}$. As shown in Equation 11, we constructed a model with a set of parameters μ_i 's, to derive the probability of an answer being a good quality answer given its feature values $f_i(a, y)$'s. The parameters μ_i 's were obtained by maximum entropy parameter estimation.

$$p(y|a) = \frac{1}{Z(a)} \exp \left[\sum_{i=1}^9 \mu_i f_i(a, y) \right] \quad (11)$$

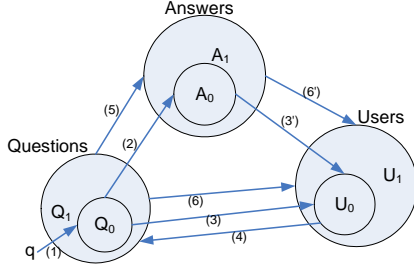


Figure 3: Steps of obtaining the QA database from a test question q .

In Equation 11, $Z(a)$ is the normalization factor, $f_i(a, y)$ is the a 's i th raw feature value for monotonic feature or converted feature value for non-monotonic feature[11]. We used Zhang Le's maximum entropy toolkit⁵ for the experiment.

Finally, Equation 12 scores a using NT method, where $qscore_{nt}(a)$ is the output of the quality predictor ($p(y|a)$).

$$score(q, a) = rscore(q, a) \cdot qscore_{nt}(a) \quad (12)$$

5.3 Dataset

We randomly selected 50 popular test questions in the *computer and internet* domain for our experiments. A user question was considered popular if Yahoo! Answers' search engine returned more than 10 other relevant questions in the top 20 questions returned, when querying the user question.

The best answers of the top 20 questions that Yahoo! Answers (BasicYA(s+c+b)) returned for each of the 50 test questions are manually judged by a group of annotators (refer to Section 5.4). The annotators assign quality label, $quality \in \{good, bad\}$, for each answer. An answer is considered in good quality if it is informative, useful, objective, sincere, readable, relevant and correct to the associated question[13].

The above 1000 quality-labeled answers were also used as training data required by NT method. To evaluate NT, we performed 5-fold Cross Validation. At each fold, we tested the performance of 10 test questions using 800 labeled answers from the remaining 40 questions as training.

We further divided the 50 test questions into Cat A and Cat B. Among the 50 test questions, 23 belong to Cat A and 27 to Cat B.

Cat A (Test questions with poor quality best answers)

These are test questions for which Yahoo! Answers returned not less than 4 of bad quality best answers from those of the top 10 questions returned.

Cat B (Test questions with good quality best answers)

These are test questions for which Yahoo! Answers returned less than 4 bad quality best answers from those of the top 10 questions returned.

The choice of 4 as threshold is empirically determined to divide the 50 questions into Cat A and Cat B of similar sizes. To obtain a database of question and answers (QA database) for our experiments, we crawled the Yahoo! Answers portal, obtaining 95,368 questions from the *Computer and Internet* category. These questions have 537,491 answers and altogether they involve 238,178 users.

Table 2: Dataset statistics

Num of questions	95,368
Num of answers	537,491
Num of users	238,178
Num of answerers	173,383
Num of askers	88,951
Num of both answerer-askers	24,156
Num of answerer-only	149,227
Num of asker-only	64,795
Average number of answers per question	5.650
Max number of answers per question	259
Average number of answers per answerer	3.102
Max number of answers per answerer	1,804
% answerers with 1 answer	67.74 %

As shown in Figure 3, we first submitted each of our 50 test questions to Yahoo! Answers search engine which returns top 50 questions denoted by Q_0 (Step 1). From Q_0 , we obtained all their answers, denoted by A_0 , (Step 2). We further gathered the set of users, denoted by U_0 , who asked or answered questions in Q_0 (Steps 3 and 3'). Q_1 , a set of questions that were answered by users in U_0 and have overlapping non-stopword terms with the test question q , was then obtained (Step 4). We then collected a set of answers, denoted by A_1 , for questions in Q_1 (Step 5). We finally gathered a set of users, denoted by U_1 , who have asked or answered questions in Q_1 (Steps 6 and 6').

As shown in Figure 3, it is obvious that $Q_0 \subseteq Q_1$, $A_0 \subseteq A_1$ and $U_0 \subseteq U_1$. We summarize the statistics of the QA database in Table 2.

5.4 Relevance and Quality Judgment

To obtain relevance and quality judgment, we conducted a user study that involves a team of nine annotators who are undergraduate students majoring in Computer Engineering and Electrical Engineering to provide answer labels. All of the users are familiar with the Computer and Internet domain used in our dataset.

We pooled all top 20 answers for each test question obtained by all experimented methods. Given that our experiment involves basic methods (BasicYAs and BasicQLs) as well as feature based quality-aware method (NT) and expertise based quality-aware methods (EXHITS, EXHITS_QD, EX_QD and EX_QD') with different answer options and parameter values (see Table 5), this corresponds to 8,617 question and answer pairs. We presented these question and answer pairs to our annotators and asked them to label the relevance of answers to the test question ($relevance \in \{relevant, irrelevant\}$) and the quality of answers ($quality \in \{good, bad\}$) based on the correctness, readability, usefulness, objectivity, and sincerity of the answers.

We grouped the annotators into three groups to provide three redundant relevance and quality judgments for each answer. Tables 3 and 4 show the percentages of labeled answers which two and three, respectively, of the groups of annotators are in agreement. An answer is considered relevant or good quality if at least $minSup$ users judge so. The minimum support $minSup$ is the minimum numbers of good judgment from users an answer received to be considered as a good answer. We consider two values of $minSup$, $minSup = 2$ and $minSup = 3$. The manually labeled data is used for both training and evaluation (ground truth set). To train the NT method (see Section 5.2), we use the 1000 quality-labeled answers returned by BasicYA method with $minSup = 3$ (stricter judgment).

⁵http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html

Table 3: Percentage of labeled answers with agreement of any two redundant judgments

relevance	Group1	Group2	Group3
Group1	100.00%	91.64%	91.58%
Group2	91.64%	100.00%	87.64%
Group3	91.58%	87.64%	100.00%

quality	Group1	Group2	Group3
Group1	100.00%	90.00%	90.00%
Group2	90.00%	100.00%	87.12%
Group3	90.00%	87.12%	100.00%

Table 4: Percentage of labeled answers with agreement of the three redundant judgments

	Group1,2,3
relevance	86.91%
quality	84.20%

5.5 Evaluation of QA Methods

We experimented all the four expertise based methods (EXHITS, EXHITS.QD, EX_QD and EX_QD') and compared them with the three baselines (BasicYA, BasicQL and NT). We summarize the methods experimented in Table 5, where *s*, *c* and *b* refer to subject, content and best answers of questions respectively, while *QL* refers to query likelihood relevance model. We used *QL* as relevance model for NT and all the expertise based methods. We will justify our choice based on our experimental results in Section 6.1.

Best Answers vs. All Answers Options. Each quality-aware method can consider using all answers or best answers as answer candidates. We differentiate the methods by giving an asterisk (*) postfix to methods using all answers. The methods without asterisk (*) postfix use best answers only.

The top 20 of the ranked answers of each methods were manually judged in terms of their relevance and quality as mentioned in Section 5.4. To evaluate the accuracy of the methods, we use the following evaluation metrics:

Precision of quality at top k ($P_{q@k}$) the proportion of good quality answers within top k answers returned

Precision of relevance at top k ($P_{r@k}$) the proportion of relevant answers within top k answers returned

Overall precision at top k ($P@k$) the proportion of both good quality and relevant answers within top k answers returned

We obtained the average value of each metric for Cat A and Cat B test questions at $k = 5, 10$ and 20 . By separating the evaluation of the two categories, we want to examine the performance of QA methods for questions where Yahoo! Answers performs poorly or well.

$P@k$ captures the overall performance while $P_{q@k}$ and $P_{r@k}$ measure quality and relevance of answers respectively. The latter two metrics are useful as relevant answers are not necessarily in good quality, and good quality answers are not necessarily relevant. However, to save space, we do not always report $P_{r@k}$ of all methods as it can be inferred by $P_{q@k}$ and P .

6. EXPERIMENTAL RESULTS

6.1 Comparing Basic and NT Methods

In Table 6, we show the precision of quality and the overall precision at top 5, 10 and 20 of BasicYA, BasicQL, NT(using

Table 5: Summary of methods

Method	Search options	Relevance model	Parameter settings
BasicYA	s+c+b and s+c	Yahoo! Answers	N/A
BasicQL	s+c and s	QL	N/A
NT	s	QL	N/A
EXHITS	s	QL	$\sigma = 0.8, 1$
EXHITS.QD	s	QL	$\sigma = 0.8, 1$
EX_QD	s	QL	$\sigma = 0.8, 1$
EX_QD'	s	QL	$\sigma = 0.8, 1$

best answers only) and NT*(using all answers) methods for both Cats A and B questions, evaluated by using our ground truth set with $minSup = 2$.

Quality and Overall Performance of Basic Methods. As shown in Table 6, answer quality affects the overall performance. Both BasicYA and BasicQL have poor overall precision for Cat A questions compared to that for Cat B questions. While the precision of relevance of both categories are comparable, the Cat A questions suffer from very low precision of quality compared to Cat B questions. This result is not surprising for BasicYA at $k = 10$, since Cat A test questions are expected to have $P_{q@10} \leq 0.6$. Such a question selection criteria clearly affects the quality performance of all Basic Methods.

Search Options and Answer Relevance Models of Basic Methods. Table 6 shows that in terms of relevance, BasicYA (s+c) gave the worst $P_{r@10}$ and $P_{r@20}$ among the basic methods. On the other hand, it gave the best $P_{r@5}$. This is because searching by question subject and content in BasicYA(s+c) often returns very few similar questions (e.g. ≤ 10) compared to Basic Methods using other search options. Nevertheless, these questions are very similar to the test question leading to relatively high $P_{r@5}$ but low $P_{r@10}$ and $P_{r@20}$.

BasicQL(s) vs. Other Basic Methods. BasicQL(s) almost consistently outperformed BasicQL (s+c) and BasicYA (s+c+b). It is much better than BasicYA(s+c) except for $P@5$. This shows that search option matters in relevance performance. We did not perform the comparison between BasicQL (s) and BasicYA (s) since Yahoo! Answers does not provide the option to search by question subject only. Nonetheless, we feel that even if this option exists, the relevance performance will not be very different from that of BasicYA (s+c).

With these findings and considering that Yahoo! Answers only returns result ranks but not relevance scores, we decided to use *QL* and question subject as our search option (as adopted by BasicQL(s)) for $rscore(q, q_i)$ and $rscore(q, a_{ij})$ to be used in all quality-aware methods.

Basic vs. NT/NT*. We now examine the difference between quality-aware non-expertise based method (NT) and Basic methods. For Cat A questions, as shown in Table 6, we observe that NT and NT* managed to improve both the quality precision and overall precision over BasicQL(s). NT (NT*) improves the overall performance of BasicQL by 14.0%, 7.5% and 14.9% (21.7%, 11.4% and 16.9%) at top 5, 10 and 20 respectively. For most of the time, the improvement is statistically significant with sign-test.

For Cat B questions however, both NT and NT* did not necessarily outperform BasicQL. At top 5 and 10, NT did poorly in overall performance compared to that of BasicQL. The same observation also holds for NT* at top 10 but at a lesser degree. We believe that the performance of BasicQL

Table 6: Performance of Basics, NT and NT*⁶, evaluated using ground truth set with $minSup = 2$.

Relevance	Cat A			Cat B		
	$P_{-r@5}$	$P_{-r@10}$	$P_{-r@20}$	$P_{-r@5}$	$P_{-r@10}$	$P_{-r@20}$
BasicYA(s+c)	0.887	0.835	0.613	0.948	0.896	0.728
BasicYA(s+c+b)	0.870	0.848	0.774	0.859	0.859	0.804
BasicQL(s+c)	0.896	0.865	0.828	0.904	0.881	0.802
BasicQL(s)	0.843	0.865	0.835	0.933	0.900	0.828
NT(s)	0.887	0.865	0.839	0.881	0.878	0.813
NT*(s)	<u>0.913^{††}</u>	<u>0.922^{††}</u>	<u>0.920^{††}</u>	<u>0.978^{††}</u>	<u>0.952^{††}</u>	<u>0.915^{††}</u>

Quality	Cat A			Cat B		
	$P_{-q@5}$	$P_{-q@10}$	$P_{-q@20}$	$P_{-q@5}$	$P_{-q@10}$	$P_{-q@20}$
BasicYA(s+c)	0.574	0.496	0.385	0.815	0.789	0.620
BasicYA(s+c+b)	0.565	0.509	0.487	0.830	0.819	0.737
BasicQL(s+c)	0.583	0.600	0.589	0.778	0.752	0.731
BasicQL(s)	0.583	0.617	0.578	0.815	0.811	0.737
NT(s)	0.678	0.665 [†]	0.657 [†]	0.741	0.763	0.756
NT*(s)	<u>0.739^{††}</u>	<u>0.674[†]</u>	0.607	0.807	0.767	0.744

Overall	Cat A			Cat B		
	$P@5$	$P@10$	$P@20$	$P@5$	$P@10$	$P@20$
BasicYA(s+c)	0.539	0.448	0.348	0.778	0.733	0.561
BasicYA(s+c+b)	0.513	0.448	0.415	0.726	0.711	0.606
BasicQL(s+c)	0.565	0.548	0.502	0.719	0.648	0.591
BasicQL(s)	0.557	0.570	0.496	0.748	0.726	0.615
NT(s)	0.635 [†]	0.613	0.570 [†]	0.630	0.656	0.628
NT*(s)	<u>0.678[†]</u>	0.635 [†]	<u>0.580[†]</u>	0.785	0.726	0.685

for Cat B questions was already so good that it is more difficult for NT or NT* to further improve the performance.

For both Cats A and B, NT* consistently yields better overall precision than NT. This was due to NT*'s significantly better precision of relevance. This suggests that considering all answers may be a better option than considering best answers only. We will re-examine the answer option issue again for other quality-aware methods in Section 6.2.

6.2 Performance of Quality-Aware Expertise Based Methods

We show the performance of BasicQL, NT and all the expertise based methods (EXHITS, EXHITS_QD, EX_QD and EX_QD') for both Cats A and B questions in Table 7 (for best answers only) and Table 8 (for all answers). All the results presented in the two tables were obtained after evaluating the methods using our ground truth set with $minSup = 2$. We shall first discuss the results for the best answers only presented in Table 7 and then compare this with the results for all answers presented in Table 8.

EXHITS and EXHITS_QD can be implemented using iterative computation. The terminating condition for this iterative computation is when ϵ , the difference in all value changes from the previous iteration to the next, is sufficiently small. Using $\epsilon = 10^{-30}$, both EXHITS and EXHITS_QD converged with no more than 175 iterations within at most 173 seconds for computing expertise on a Pentium 4 computer 3.4 GHz (2 CPU) with 1 GB of RAM. Similar to HITS, the expertise computations of EXHITS and EXHITS_QD depend on the connections of askers and answerers through asking and answering questions. In our experiments, at least 82.26% of the askers and answerers are connected, and large majority of them have non-zero expertise scores.

Importance of Asking Expertise. To study the importance of answerer's asking expertise in calibrating the quality of answers, we experimented EXHITS, EXHITS_QD, EX_QD, EX_QD' with $\sigma = 0.8$ (using asking expertise) and

$\sigma = 1$ (ignoring asking expertise). We chose $\sigma = 0.8$ empirically with the intuition that answering expertise is more important than asking expertise in determining the quality of answers.

As shown in Table 7, expertise based methods with $\sigma = 0.8$ yield almost consistently better answers than those with $\sigma = 1$ especially for Cat A questions. We further notice that the difference in performance at top 10 and top 20 is more significant compared to that at top 5. This suggests that answering expertise is indeed more important than asking expertise. Answering expertise alone is good enough to obtain a few good answers at the top ranks. However if we would like to have more good answers, we should use answerer's asking expertise in addition to their answering expertise. More good answers can be useful in applications where we need to synthesize a comprehensive answer from raw answers.

Expertise Based Methods. Among EXHITS, EXHITS_QD, EX_QD and EX_QD', we observe that in general, question dependent is better than question independent ones across different categories and evaluation metrics. This supports the idea that the ability of a user in providing good answers varies with question topic.

Furthermore, as shown in Table 7 we notice that EXHITS, EXHITS_QD did not outperform EX_QD, EX_QD' in overall performance. This suggests that peer expertise dependency is not important in determining the overall goodness of answers when compared to the topic of the user question. One possible explanation to this is that the assumption that good answerers tend to answer good askers' questions does not hold in community QA portals such as Yahoo! Answers.

Expertise Based vs. NT. As shown in Table 7, our proposed methods EX_QD and EX_QD' are the best methods among all regardless of categories and evaluation metrics. Moreover, these methods improvement over NT is statistically significant for almost all of the evaluation metrics. For Cat A questions, EX_QD and EX_QD' improved the overall precision of NT by 10.87%, 12.72% and 12.46% at top 5, 10 and 20 respectively. These methods' improvement over BasicQL for Cat A questions is even more significant.

Furthermore, for Cat B questions, unlike NT and NT* that did not always perform better than BasicQL, EX_QD and EX_QD' consistently outperformed BasicQL for these questions.

Best Answer Candidates vs. All Answer Candidates. As shown in Table 8, expertise based methods using all answers option (EXHITS*, EXHITS_QD*, EX_QD* and EX_QD'*) show similar behavior as those using best answers option. Firstly, the asking expertise is also important to obtain more good answers. Secondly, EX_QD* and EX_QD'* also performed the best most of the time when using all answers option.

By comparing Tables 7 and 8, we further observe that all answers option is indeed a better answer option than best answers option. While all answers option did not improve much in P_{-q} , it helped all methods to significantly improve the overall performance. The performance gain is derived from better relevance precision. Interestingly, many of the relevant non-best answers turn out to be good quality. By

^{6†} and ^{††} indicate statistically significant improvement of NT or NT* methods over BasicQL with a 95% and 97.5% confidence levels respectively, according to sign-test. The best result for each evaluation metric is underlined

Table 7: Performance of BasicQL, NT and expertise based methods for best answers only⁷, evaluated using ground truth set with $minSup = 2$.

Quality	σ	Cat A			Cat B		
		$P_{-q@5}$	$P_{-q@10}$	$P_{-q@20}$	$P_{-q@5}$	$P_{-q@10}$	$P_{-q@20}$
BasicQL		0.583	0.617	0.578	0.815	0.811	0.737
NT		0.678 ^{††}	0.665 [†]	0.657	0.741	0.763	0.756
EXHITS	0.8	0.765^{††}	0.761^{††}	0.728 ^{††}	0.793	0.789	0.744
	1	0.748 ^{††}	0.726 [†]	0.743^{††}	0.807	0.789	0.746
EXHITS _{-QD}	0.8	0.748^{††}	0.735[†]	0.728 ^{††}	0.822	0.789	0.807
	1	0.748^{††}	0.735[†]	0.743^{††}	0.756	0.774	0.772
EX_QD	0.8	0.756^{††*}	0.770^{††**}	0.772^{††**}	0.815	0.826*	0.828 ^{†*}
	1	0.678 ^{††}	0.726 [†]	0.757 ^{††*}	0.785	0.819	0.833^{††**}
EX_QD'	0.8	0.756^{††*}	0.770^{††**}	0.772^{††**}	0.815	0.826*	0.828 ^{†*}
	1	0.678 ^{††}	0.726 [†]	0.757 ^{††*}	0.785	0.819	0.833^{††**}

Overall	σ	Cat A			Cat B		
		$P@5$	$P@10$	$P@20$	$P@5$	$P@10$	$P@20$
BasicQL		0.557	0.570	0.496	0.748	0.726	0.615
NT		0.635 [†]	0.613	0.570	0.630	0.656	0.628
EXHITS	0.8	0.600	0.570	0.450	0.615	0.522	0.448
	1	0.617	0.574	0.454	0.630	0.522	0.452
EXHITS _{-QD}	0.8	0.635	0.609	0.563 [†]	0.667	0.670	0.583
	1	0.609	0.604	0.511	0.630	0.596	0.519
EX_QD	0.8	0.704[†]	0.691^{†*}	0.641^{†*}	0.768**	0.726*	0.654
	1	0.661	0.652 [†]	0.637	0.726	0.722	0.672^{†*}
EX_QD'	0.8	0.704[†]	0.691^{†*}	0.641^{†*}	0.756**	0.726*	0.654
	1	0.661	0.652 [†]	0.637	0.726	0.722	0.672^{†*}

allowing only one best answer selected for each question, many equally good or even better answers have been neglected by the methods using best answer option. For Cat A questions, EX_QD* and EX_QD'* significantly improved the overall precision of NT* by 7.67%, 9.61% and 18.79% at top 5, 10 and 20 respectively. For Cat B questions, EX_QD* and EX_QD'* also improved the overall performance of both BasicQL and NT* more than what EX_QD or EX_QD' had done to BasicQL and NT.

Consistency of performance with different strictness of judgment. We also present the performance of BasicQL, NT and all the expertise based methods for both Cats A and B questions evaluated using stricter judgment (ground truth set with $minSup = 3$) in Table 9 (for best answers only) and Table 10 (for all answers). With stricter judgment, as expected, slight degradation of performance is observed for all the experimented methods. Nevertheless, the relative performance among different methods remained mostly unchanged. Generally, expertise based methods, especially question dependent methods, still significantly outperformed the basic and NT methods when evaluated with stricter judgment. This affirms the strength of our methods as compared to other competitors. To conserve space, the tables show the overall performance only.

7. CONCLUSIONS AND FUTURE WORK

In this paper, we develop collaborative QA methods over a community QA portal such as Yahoo! Answers. We show that quality, other than relevance, is an important criteria for selecting answers to be returned. Based on our proposed quality-aware QA framework, we have introduced

^{7†}(*) and ^{††}(**) indicate statistically significant improvement of quality-aware expertise based methods towards BasicQL (NT or NT*) with confidence levels of 95% and 97.5% respectively, according to sign-test. For each expertise based method, we emphasize in bold the best result among two different σ values ($\sigma = 0.8$ and $\sigma = 1$).

Table 8: Performance of BasicQL, NT and expertise based methods for all answers⁷, evaluated using ground truth set with $minSup = 2$.

Quality	σ	Cat A			Cat B		
		$P_{-q@5}$	$P_{-q@10}$	$P_{-q@20}$	$P_{-q@5}$	$P_{-q@10}$	$P_{-q@20}$
BasicQL		0.583	0.617	0.578	0.815	0.811	0.737
NT*(s)		0.739 ^{††}	0.674 [†]	0.607	0.807	0.767	0.744
EXHITS*	0.8	0.713[†]	0.683	0.702[†]	0.815	0.815	0.811
	1	0.670	0.665	0.685 [†]	0.704	0.715	0.731
EXHITS _{-QD} *	0.8	0.765^{††}	0.713^{†*}	0.683[†]	0.859	0.815	0.781
	1	0.739 ^{††}	0.696	0.670 [†]	0.859	0.815	0.778
EX_QD*	0.8	0.774^{††}	0.717^{†*}	0.713^{††*}	0.859	0.822*	0.811^{†*}
	1	0.765 ^{††}	0.713 ^{†*}	0.689 [†]	0.837	0.811	0.811^{†*}
EX_QD'*	0.8	0.774^{††}	0.722^{†*}	0.713^{††*}	0.859	0.822*	0.811^{†*}
	1	0.765 ^{††}	0.713 ^{†*}	0.689 [†]	0.837	0.811	0.811^{†*}

Overall	σ	Cat A			Cat B		
		$P@5$	$P@10$	$P@20$	$P@5$	$P@10$	$P@20$
BasicQL		0.557	0.570	0.496	0.748	0.726	0.615
NT*		0.678 [†]	0.635 [†]	0.580 [†]	0.785	0.726	0.685
EXHITS*	0.8	0.661	0.617	0.624	0.756	0.737	0.678
	1	0.609	0.583	0.576	0.615	0.622	0.593
EXHITS _{-QD} *	0.8	0.730[†]	0.665[†]	0.689^{††*}	0.822[†]	0.748	0.670
	1	0.713	0.643	0.609	0.822[†]	0.748	0.663
EX_QD*	0.8	0.730[†]	0.696^{†*}	0.689^{††*}	0.822[†]	0.785^{†*}	0.772^{†*}
	1	0.722	0.691 [†]	0.663 [†]	0.807	0.781	0.763 ^{†*}
EX_QD'*	0.8	0.730[†]	0.696^{†*}	0.689^{††*}	0.822[†]	0.785^{†*}	0.772^{†*}
	1	0.722	0.691 [†]	0.663 [†]	0.807	0.781	0.763 ^{†*}

Table 9: Overall performance of BasicQL, NT and expertise based methods for best answers only⁷, evaluated using ground truth set with $minSup = 3$.

Overall: P	σ	Cat A			Cat B		
		$P@5$	$P@10$	$P@20$	$P@5$	$P@10$	$P@20$
BasicQL		0.557	0.561	0.463	0.674	0.685	0.609
NT		0.600	0.578	0.537	0.600	0.630	0.613
EXHITS	0.8	0.574	0.543	0.428	0.600	0.500	0.424
	1	0.530	0.509	0.443	0.548	0.500	0.411
EXHITS _{-QD}	0.8	0.617	0.574	0.454	0.630	0.522	0.452
	1	0.600	0.570	0.450	0.615	0.522	0.448
EX_QD	0.8	0.696^{†*}	0.683^{†*}	0.628^{††*}	0.733^{†*}	0.700	0.654^{†*}
	1	0.626	0.626	0.607 [†]	0.726 [†]	0.696	0.633 [†]
EX_QD'	0.8	0.696^{†*}	0.683^{†*}	0.628^{††*}	0.733^{†*}	0.700	0.654^{†*}
	1	0.626	0.626	0.607 [†]	0.726 [†]	0.696	0.633 [†]

Table 10: Overall performance of BasicQL, NT and expertise based methods for all answers⁷, evaluated using ground truth set with $minSup = 3$.

Overall: P	σ	Cat A			Cat B		
		$P@5$	$P@10$	$P@20$	$P@5$	$P@10$	$P@20$
BasicQL		0.557	0.561	0.463	0.674	0.685	0.609
NT*		0.670 [†]	0.635 [†]	0.580 [†]	0.763	0.715	0.674
EXHITS*	0.8	0.635[†]	0.578	0.585	0.689	0.700	0.650
	1	0.496	0.478	0.504	0.591	0.548	0.526
EXHITS _{-QD} *	0.8	0.670[†]	0.600	0.565	0.778	0.715	0.624
	1	0.591	0.548	0.526	0.719	0.667	0.570
EX_QD*	0.8	0.687[†]	0.661[†]	0.641^{††*}	0.807[†]	0.741^{†*}	0.731^{†*}
	1	0.678 [†]	0.648 [†]	0.620 [†]	0.770	0.737 [†]	0.722 [†]
EX_QD'*	0.8	0.713^{††*}	0.652[†]	0.628^{††*}	0.785[†]	0.741^{†*}	0.722[†]
	1	0.678 [†]	0.648 [†]	0.620 [†]	0.770	0.737 [†]	0.722[†]

several QA methods including NT, EXHITS, EXHITS_QD, EX_QD, and EX_QD' that use both answer quality and answer relevance. NT uses answer features to determine answer quality, while the other four methods use different expertise models to determine answer quality from the answerer's expertise.

Our experiments on a collection of Yahoo! Answers questions and answers have shown that the quality-aware QA methods (NT, EXHITS, EXHITS_QD, EX_QD, EX_QD') enjoy better performance than the non-quality aware ones. Among the former, EX_QD and EX_QD', the two methods using question-dependent expertise (that covers both answering and asking expertise) have the best performance. Our results also show that all answers of existing questions should be considered as this increases the pool of relevant and quality answers.

Based on our results so far, we believe that collaborative question answering on a community QA portal has great potential in returning good answers especially when there is a large pool of existing questions and answers. The overall performance reported is much better than the traditional QA performance.

Looking ahead, there are many interesting directions to pursue in this track of research. We can expand our experiments to include questions that are non-popular which adds more difficulty to the QA task. We also plan to extend and evaluate our methods on questions of other domains (instead of *Computer and Internet* category). While the notion of best answer applies well to the *Computer and Internet* category, it may be less relevant to questions from other categories which expect answers tailored to different personal preferences. In these cases, the most personalized answers instead of best answers may be more appropriate. For the expertise based methods, different ways to combine relevance and quality scores (as opposed to score product) and the choice of σ can also be further investigated.

8. REFERENCES

- [1] E. Agichtein, C. Castillo, and D. Donato. Finding High-Quality Content in Social Media. In *WSDM*, 2008.
- [2] E. Agichtein, S. Lawrence, and L. Gravano. Learning to Find Answers to Questions on the Web. *TOIT*, 4(2):129–162, May 2004.
- [3] K. Balog, L. Azzopardi, and M. de Rijke. Formal Models for Expert Finding in Enterprise Corpora. In *SIGIR*, 2006.
- [4] J. Bian, Y. Liu, E. Agichtein, and H. Zha. Finding the Right Facts in the Crowd: Factoid Question Answering over Social Media. In *WWW*, 2008.
- [5] R. D. Burke, K. J. Hammond, V. A. Kulyukin, S. L. Lytinen, N. Tomuro, and S. Schoenberg. Question Answering from Frequently Asked Question Files: Experiences with the FAQ Finder System. In *Technical report*, 1997.
- [6] C. S. Campbell, P. P. Maglio, A. Cozzi, and B. Dom. Expertise Identification using Email Communications. In *CIKM*, 2003.
- [7] R. D'Amore. Expertise Community Detection. In *SIGIR*, 2004.
- [8] H. T. Dang, J. Lin, and D. Kelly. Overview of the TREC 2006 Question Answering Track. In *TREC*, 2006.
- [9] Z. Gyöngyi, G. Koutrika, J. Pederson, and H. Garcia-Molina. Questioning Yahoo! Answers. In *WWW*, 2008.
- [10] S. Harabagiu, F. Lacatusu, and A. Hickl. Answering Complex Questions with Random Walk Models. In *SIGIR*, 2006.
- [11] J.-N. Hwang, S.-R. Lay, and A. Lippman. Nonparametric Multivariate Density Estimation: A Comparative Study. *IEEE Transactions of Signal Processing*, 42(10):2795–2810, 1994.
- [12] J. Jeon, B. Croft, and J.-H. Lee. Finding Similar Questions in Large Question and Answer Archives. In *CIKM*, 2005.
- [13] J. Jeon, B. Croft, J.-H. Lee, and S. Park. A Framework to Predict the Quality of Answers with Non-textual Features. In *SIGIR*, 2006.
- [14] V. Jijkoun and M. de Rijke. Retrieving Answers from Frequently Asked Questions Pages on the Web. In *CIKM*, 2005.
- [15] P. Jurczyk and E. Agichtein. Discovering Authorities in Question Answer Communities by Using Link Analysis. In *CIKM*, 2007.
- [16] P. Jurczyk and E. Agichtein. HITS on Question Answer Portals: Exploration of Link Analysis for Author Ranking. In *SIGIR*, 2007.
- [17] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, September 1999.
- [18] A. L. Berger, S. A. D. Pietra, and V. J. D. Pietra. A Maximum Entropy Approach to Natural Language Processing. *Computational Linguistics*, 22(1):39–71, 1996.
- [19] G. Luo, C. Tang, and Y. li Tian. Answering Relationship Queries on the Web. In *WWW*, 2007.
- [20] D. Moldovan, S. Harabagiu, R. Girju, P. Morarescu, F. Lacatusu, A. Novischi, A. Badulescu, and O. Bolohan. LCC Tools for Question Answering. In *TREC*, 2002.
- [21] J. M. Ponte and W. B. Croft. A Language Modeling approach to Information Retrieval. In *SIGIR*, 1998.
- [22] E. Snieders. Automated FAQ Answering: Continued Experience with Shallow Language Understanding. In *AAAI*, 1999.
- [23] I. Soboroff, A. P. de Vries, and N. Craswell. Overview of the TREC 2006 Enterprise Track. In *TREC*, 2006.
- [24] Q. Su, D. Pavlov, J.-H. Chow, and W. C. Baker. Internet-Scale Collection of Human-Reviewed Data. In *WWW*, 2007.
- [25] Xiaoyong and W. B. Croft. Finding Experts in Community-Based Question-Answering Services. In *CIKM*, 2005.
- [26] C. Zhai and J. Lafferty. A Study of Smoothing Methods for Language Models Applied to Information Retrieval. *ACM TOIS*, 22(2):179–214, April 2004.
- [27] J. Zhang, M. S. Ackerman, and L. Adamic. Expertise Networks in Online Communities: Structure and Algorithms. In *WWW*, 2007.